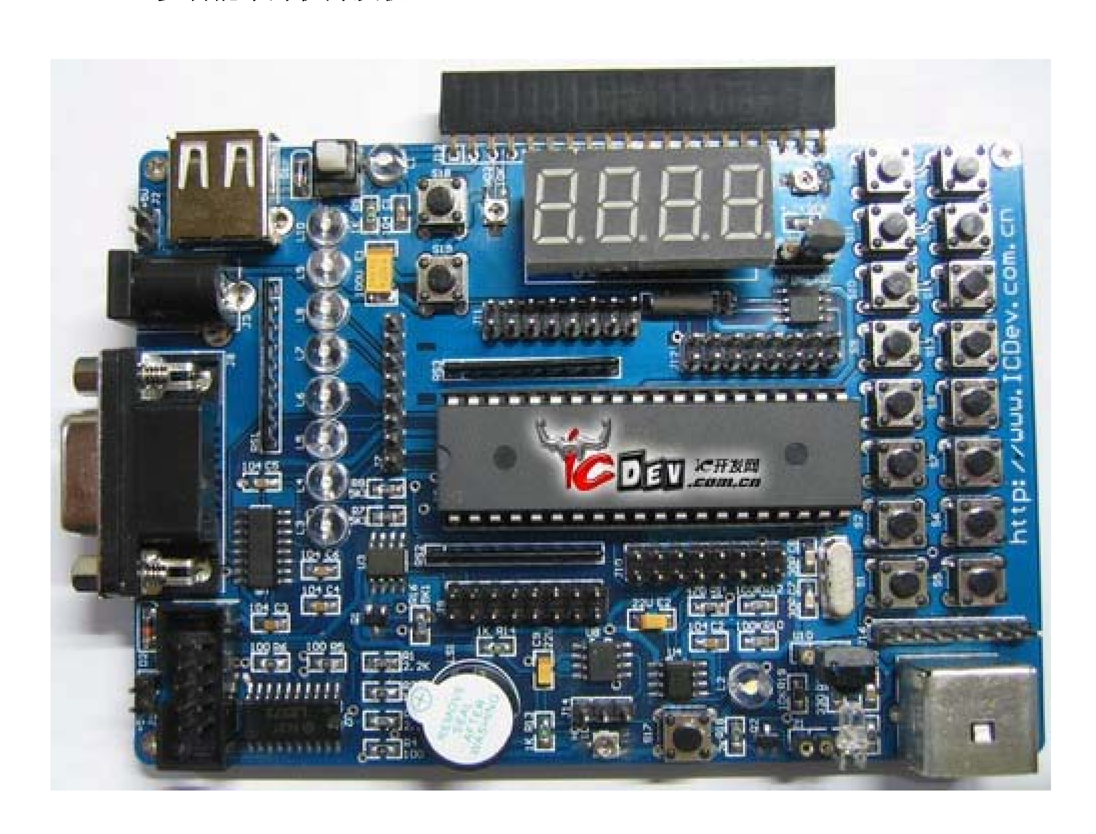
AT89S52 单片机开发板实验手册

概述：

AT89S52 多功能单片机开发板



特点：

1. 板载资源丰富,常见的控制对象基本已经包括.
2. 采用在系统方式(ISP)编程,通过下载电路下载程序,不用烧录器即可下载程序,调试方便.
3. 可由直流稳压电源或通过 USB 数据线供电.按键保持的电源开关,连接线路时只用按一下即可断开电源.扩展了若干个电源接口,板上电源可以外接给其他电路板使用.
4. 所有板上资源均用排针引出,可方便用 ARM,DSP 或其他控制器控制.
5. 红外发射/接收 ,PS/2 接口接入键盘,时钟芯片,数字温度计测温等模块的程序编成了子函数,已经模块化,工程应用需要时可以直接调用.
6. 配套光盘提供程序,常用软件(编译软件,烧录软件,字模提取,串口上位机软件,C 语言所有库函数等),视频教程(主要为计算机屏幕操作的录像),开发板用户手册,原理图,器件 Datasheet 等.配套提供的所有程序均可直接运行,注释已经尽量详细,能满足从单片机入门到进阶的需要.
7. 基于以上各模块,提供了综合应用的实例,即:可通过 PC 机键盘或通过红外发射进行控制的电子万年历等.(具体见实验介绍).

本开发板含有如下功能模块：

01. 流水灯 (控制 8 个超亮 LED 闪烁)
02. 方波,PWM 信号输出
03. 按键中断
04. 四位数码管动态显示
05. 计数脉冲信号并显示
06. 报警与音乐演奏
07. 4×4 键盘扩展
08. 看门狗+上电自动复位+手动复位+电源监控
09. 128*64 点阵式液晶显示 (可显示画面和文字等)
10. RS232 串行通讯 (单片机通过串口与 PC 机进行通讯,板上数码管和上位机软件均可显示通讯的数据,提供上位机软件)
11. 红外信号发射与接收 (单片机控制红外二极管发射红外信号并接收信号,也可以通过遥控器发射,单片机控制接收信号并处理)
12. 温度测量
13. I2C 接口的 E2PROM (AT24C08,8K 容量,可用于断电时存储数据等场合,单片机软件模拟 I2C 总线协议与之通讯)
14. 时钟芯片 (扩展时钟芯片以获取年/月/日,当前时间,星期等信息)
15. PS2 接口 (可外接 PC 机的键盘做系统的输入)
16. 可通过键盘/红外遥控控制的电子万年历 (液晶屏显示年/月/日,当前时间,星期,闹铃定闹的时间,倒计时状态等;可以修改闹铃时间和倒计时的起始时间;可显示环境温度;可通过 PS/2 接口接入键盘或通过红外发射来校正时间,开/关闹铃等.)

实验注意事项：

1. 板上扩展有两个电源接口(接线柱),正负极已经标出,外接时极性不能接反.
2. 通电时,最好不要用手拿着电路板,防止短路；接线时最好断开电源(开关为J1).
3. 调试出现故障注意查看复位电路是否接入(最好直接接上上电自动复位电路(即J14中将上电自动复位端和复位端用短路冒短接,接好后还可以通过S17手动复位),看门狗复位可以在后面再接入).
4. 下载程序时单片机P1.5,P1.6,P1.7三个脚禁止接到功能模块,否则会影响程序下载.程序比较大时,5K,6K或以上的代码时,下载可能比较慢,如果下载失败,一般重试一次即可成功下载.
5. 由于并口的影响,断开电源时可能发光LED也有微弱发光,为正常现象.
6. 程序以光盘中程序目录下的为准.
7. 数码管和液晶不能同时接入系统,否则任一部分都不能正常使用.

编程软件和烧录软件等的使用见视频教程.

如有疑问可到论坛发帖.

实验目录

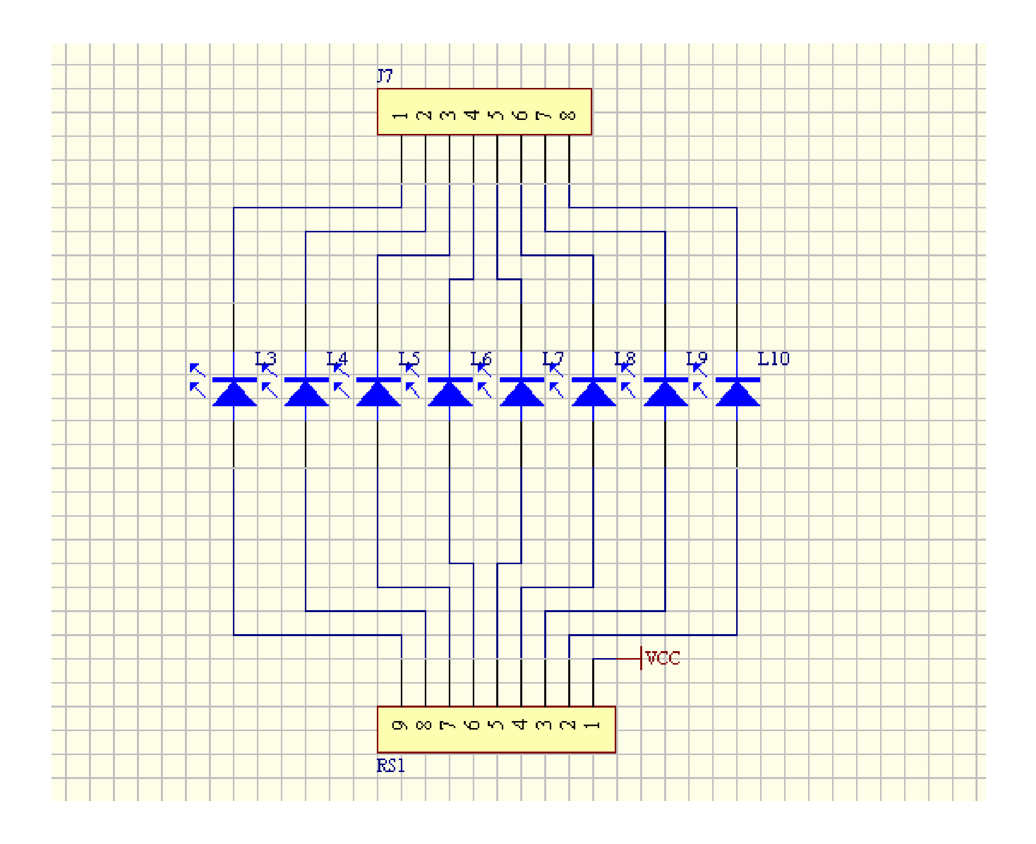
（Ⅰ）基本实验

（Ⅱ）综合实验

综合应用的实例会不断添加,敬请关注我们的网站.

（Ⅰ）基本实验

1. 流水灯



实验内容:

利用单片机 I/O 口输出高低变化的电平,控制流水灯按程序设置的功能闪烁.

接线说明

J7 接入单片机 P0 口.

注意:

单片机端口有两种操作方式,一种是写端口,如 P0=0xff; 另一种是读端口 ,如 if(P0==0xff) ,即读 I/O 口 P0 的状态,值为 0xff 时满足条件.

程序清单:

```
/******************************************************************
函数功能:   控制 8 个流水灯闪烁
接线说明:   流水灯接线柱接到 P0 口
学习内容:   单片机 IO 口(输入/输出)的应用
整理时间:   2006-10    http://www.ICDev.com.cn
******************************************************************/
```

```
#include<reg52.h>

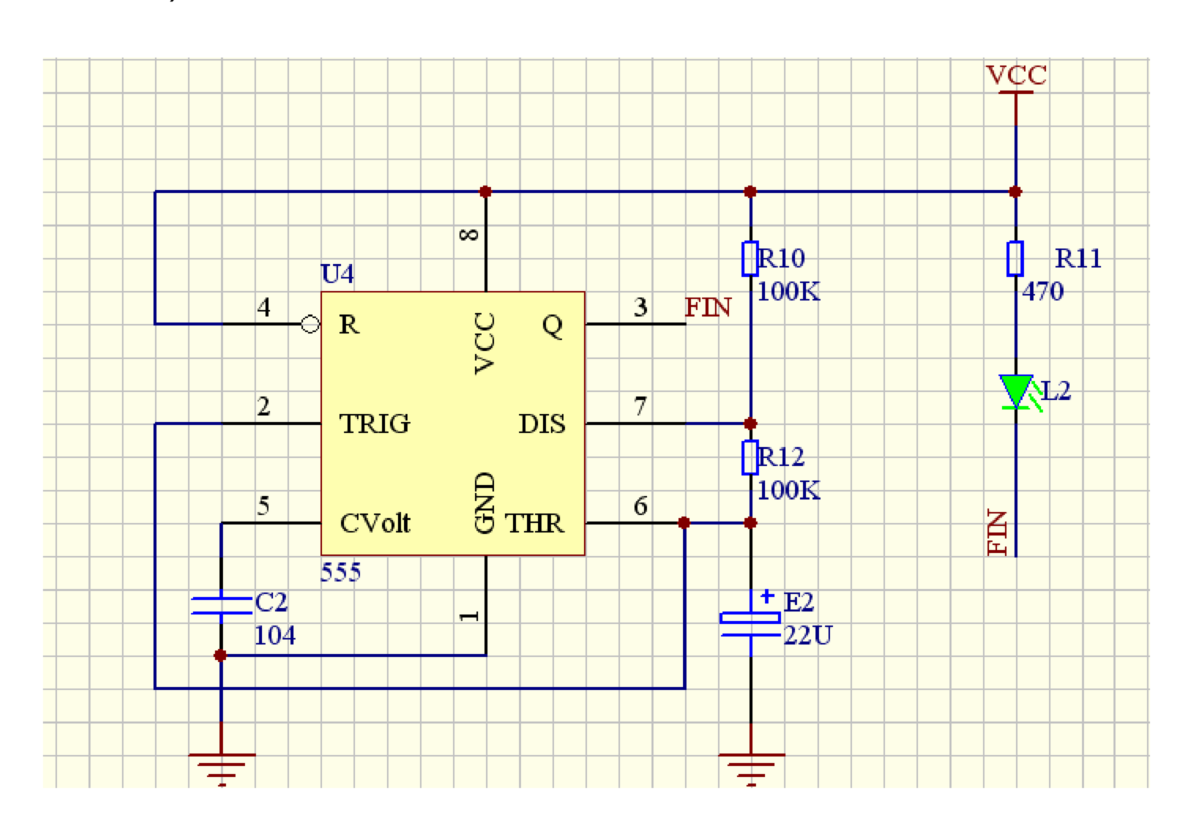
delay(int k);                                      //延时函数预定义
main()
{
   int i;
   int data1=0x7f;
   int data2=0xfe;
   while(1)
    {
       int abb=0xfe;                               //变量 abb 赋初值
       for(i=0;i<8;i++)
        {
          P0=abb;
          delay(1000);                             //延时 1 秒
          abb=abb<<1;                              //8 个灯依次点亮
        }
       P0=0xff;                                    //8 个灯均灭
       delay(1000);
       abb=0x7f;
       for(i=0;i<8;i++)
        {
          P0=abb;
          delay(1000);
          abb=abb>>1;                              //8 个灯换个方向依次点亮
        }
       P0=0xff;                                    //8 个灯均灭
       delay(1000);
       P0=0x00;                                    //8 个灯均灭
       delay(1000);
       P0=0xff;                                    //8 个灯均灭
       delay(1000);
    }
}

delay(int k)                                       //延时函数,大致延时 K 毫秒
{
      int d,n;
      for(d=0;d<k;d++)
       {
         for(n=0;n<125;n++){;}      //由经验值,12M 晶振时大概的延时时间(延时 1ms)
       }
}
```

2. 方波输出

<1>. 利用板上 NE555 组成振荡器产生方波.

占空比及周期可以通过相关电阻,电容的数值计算得出.(线路已经接好,上电后 L2 即开始闪烁)



<2>. 通过单片机定时器产生方波

通过设定定时器的定时值,计数值满时溢出产生中断,在中断服务程序中使相应管脚输出电平反相,从而产生方波信号.

接线说明:

单片机 P1.2 接到流水灯上(J7)的任意一个

程序清单:

```
/************************************************************************
函数功能:  利用内部定时器输出方波控制流水灯
接线说明:  P1.2 接到流水灯上(J7)的任意一个
学习内容:  单片机内部定时器/计数器的应用
编程思想:  单片机内部定时器最大定时值不超过一秒,   可以用累加的方式增加定时时间
整理时间:  2006-10        http://www.ICDev.com.cn
************************************************************************/
#include <reg52.h>

int   time=0;
sbit P1_2=P1^2;                    //定义控制脚
```

```
void timeout1() interrupt 1 using 2            //定时器 0 的中断服务程序
{
  if(++time==100)                              //延时 100*10ms(即 1 秒)时输出反向
   {
      P1_2=~P1_2;
      time=0;
   }
  TH0=(65536-10000)/256;                       //重装数据,延时 10ms
  TL0=(65536-10000)%256;
  TR0=1;
}

main()
{
  TMOD=TMOD&0xf0|0x01;                         //定时器 0 工作在方式 1
  EA=1;
  ET0=1;
  TH0=(65536-10000)/256;                       //延时 10ms
  TL0=(65536-10000)%256;
  TR0=1;
  while(1){;}
}
```

<3>.单片机输出 PWM 波形

输出 PWM 信号的原理与产生方波大致一样.设定一个数组,存入不同的延时值,定时器装入初值,溢出后产生中断,在中断服务取出数组中的定时值赋给定时器的数据寄存器 ，从而产生 PWM 波形.高低电平的维持时间有数组的值决定.

接线说明:

单片机 I/O 口 P1.0 接到流水灯上的任意一个.

程序清单:

```
/*********************************************************************
函数功能:   利用内部定时器输出 PWM 信号控制流水灯
接线说明:   P1.0 接到流水灯上的任意一个
学习内容:   单片机内部定时器/计数器的应用
编程思想:   输出低电平 1000 微秒，高电平 15000 微秒(12M 的晶振时),
            可应用于电力电子技术中升压斩波电路等做 IGBT 的触发信号等.
整理时间:   2006-10     http://www.ICDev.com.cn
*********************************************************************/

#include <reg52.h>
sbit P1_0=P1^0;                        //定义变量 P1_0 代表 I/O 口 P1 口的 0 脚
```

```
bit i=0;
int tab[2]={1000,15000};

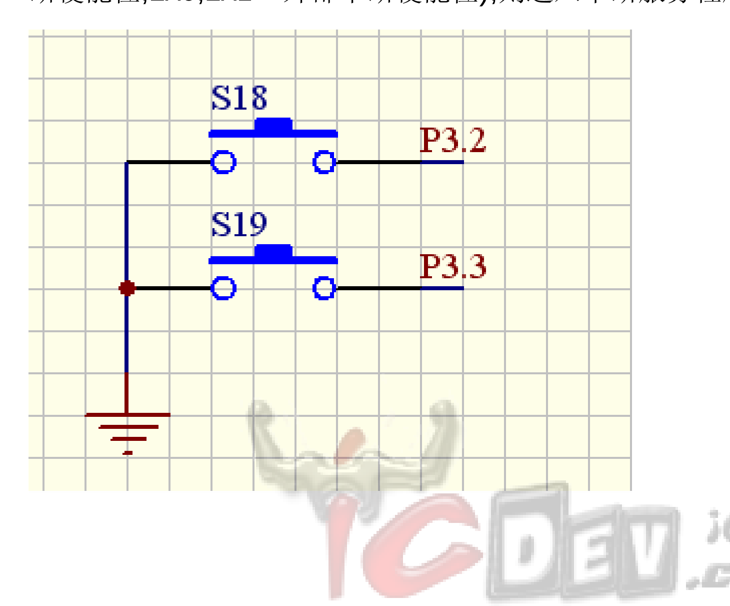
void timeserver() interrupt 3 using 3              //定时器 1 中断服务程序
{
   P1_0=~P1_0;                                     //输出取反
   i=~i;
   TH1=(65536-tab[i])/256;
   TL1=(65536-tab[i])%256;
   TR1=1;
}


main()
{
   P1_0=0;
   TMOD=TMOD&0x0f|0x10;           //不改变定时器 0 的工作状态,定时器 1 工作于方式 1
   TH1=(65536-tab[i])/256;        //定时器存入初值
   TL1=(65536-tab[i])%256;
   EA=1;                          //CPU 开中断
   TR1=1;                         //定时器 1 开始工作
   while(1){;}                    //等待定时器中断

}
```

3. 按键中断

通过设定单片机中断的形式,如下降沿产生中断或低电平产生中断.按键输入低电平到相应中断输入管脚,单片机识别出中断信号,如果 CPU 和相应外部中断使能有效(即 EA---CPU 中断使能位,EX0,EX1---外部中断使能位),则进入中断服务程序.



接线说明:

中断按键的线路已经连接好,不用另外接线. 数码管用跳线冒接入(即 AA 连接到 P0.0,BB 连接到 P0.1 依次接入,CON1...CON4 分别接到 P2.7......P2.4)

程序清单:

```
/*****************************************************************
函数功能:  利用内部中断控制数码管显示
接线说明:  数码管用跳线冒接入
学习内容:  单片机中断(外部)的应用
编程思想:  单片机识别出中断 0 或中断 1 就转向中断服务程序
整理时间:  2006-10          http://www.ICDev.com.cn
*****************************************************************/
#include <reg52.h>
bit flag1=0,flag2=0;

void service_int1() interrupt 0          //外部中断 0 的中断服务程序，0 为中断源序号，2 为
                                         //使用的寄存器(可更改)
{
   flag1=1;                              //允许数码管显示数据
}

void service_int2() interrupt 2          //外部中断 1 的中断服务程序
{
   flag2=1;                              //允许数码管显示数据
}

void delay()                             //延时程序
{
  int j;
  for(j=32400;j>0;j--){;}
}

void main()
{
   EX0=1;                                //开外部中断
   EX1=1;
   EA=1;                                 //开 CPU 中断
   P0=0x00;
   P2=0xff;
   IP=0x04;                              //设置中断优先级，外部中断 0 的级别设为最高，外部
中断 1 和其他中断同级
                                         //不设优先级则按单片机硬件决定
   while(1)
    {
```

```
    P0=0x00;
    P2=0xff;
    if(flag1==1)
     {
      P0=0xd5;                    //显示 3
      P2=0x00;
      flag1=0;
      delay();
     }
    if(flag2==1)
     {
      P0=0xdb;                    //显示 6
      P2=0x00;
      flag2=0;                    //清除显示使能
      delay();
      }
  }
}
```

4. 四位数码管动态显示

数码管每个显示数字共用数据线(8 根线),每个分别有一个使能管脚,显示数据采用隐消的编程方法,即每一位显示一段时间再轮换.

接线说明:

数码管用跳线冒接入(即 AA 连接到 P0.0,BB 连接到 P0.1 依次接入,CON1...CON4 分别接到 P2.7......P2.4).**数码管何液晶不能同时接入系统,否则任一部分都不能正常使用.**

程序清单:

```
/***********************************************************************
函数功能:   数码管数值动态显示
接线说明:   用跳线冒接入数码管
学习内容:   常用的输出显示器件的编程应用
编程思想:   用隐消的方法实现数码管的动态显示,显示部分整理成库文件,方便以后直接应用
整理时间:   2006-10      http://www.ICDev.com.cn
***********************************************************************/
#include <reg52.h>
#include <LED8888.h>                        //调用封装好的显示库函数

main()
{
 int j,k;
 k=1286;                                    //显示值,实现数值累加显示
```

```
 while(1)
 {
    for(j=1;j<70;j++)
    {
     Led(k);                                    //数码显示
    }
    k+=1;
 }
}
```

以下为 LED8888.h

```
char shuju[10]={0x5f,0x44,0x9d,0xd5,0xc6,0xd3,0xdb,0x45,0xdf,0xd7}; //数码管显示数字
0－9

void led8888_Delay(int j)                       //延时函数
{
       int m;
       for(m=0;m<j;m++)
       {
              int i;
              for(i=0; i<300; i++){;}           //约 1ms
       }
}


void Led(int k)
{
   if(k>999)
   {P2=P2&0x7f;                                 //P2.7 输出低电平，选通千位数
    P0=shuju[k/1000];                           //取千位数
    led8888_Delay(2);                           //延时
    P2=P2|0xff;                                 //销隐
   }
   if(k>99)
   {
    P2=P2&0xbf;                                 //P2.6 输出低电平，选通百位数
    P0=shuju[k%1000/100]+0x20;                  //取出百位数,点亮数码管百位后的那一点
    led8888_Delay(2);                           //延时
    P2=P2|0xff;                                 //销隐
   }
   if(k>9)
   {
    P2=P2&0xdf;                                 //P2.5 输出低电平，选通十位数
```

```
    P0=shuju[k%100/10];               //取十位数
    led8888_Delay(2);                 //延时
    P2=P2|0xff;                       //销隐
   }
  if(k>=0)
  {P2=P2&0xef;
    P0=shuju[k%10];                   //取出个位数
    led8888_Delay(2);
    P2=P2|0xff;
   }
}
```

5. 计数脉冲信号并显示

利用单片机计数器功能,计数外部输入的脉冲信号,计算得出频率值.

接线说明:

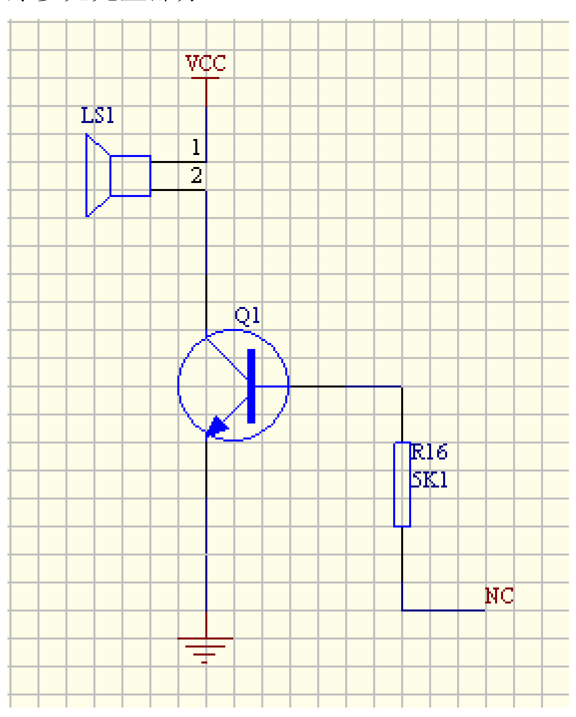
FIN 接入单片机 I/O 口 P3.5. 数码管接入

程序清单:

```
/*********************************************************************
函数功能:  计数外部(555 电路)的脉冲
接线说明:  FIN 接入单片机 I/O 口 P3.5.
学习内容:  单片机内部定时器/计数器的应用
整理时间:  2006-10        http://www.ICDev.com.cn
*********************************************************************/
#include <reg52.h>
#include <LED8888.h>
int count=0;                          //显示值赋初值
main()
{
  TMOD=0x55;                          //计数器 1 工作在方式 1
  TL1=(65536-5)%256;                  //设置初值,计数 4 次后产生定时/计数器中断
  TH1=(65536-5)/256;
  TR1=1;                              //启动计数器
  EA=1;                               //开 CPU 中断,不开中断也行,没有用到计数器中断
  ET1=1;
  while(1)
    {
      count=TL1;                      //显示当前计数值
      Led(count);
    }
}
```

6. 报警与音乐演奏

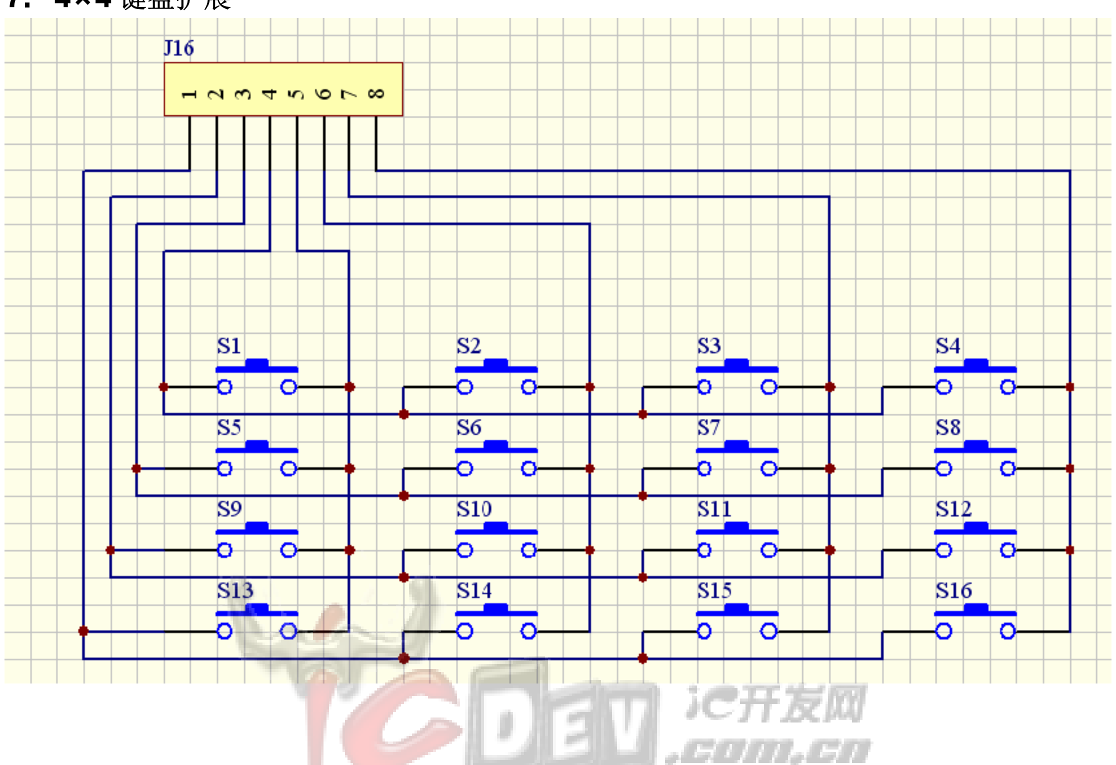
通过单片机内部定时器,输出频率改变的信号给讯响器电路,并改变节拍即可产生音乐.程序参见光盘部分.



接线说明:

NC 接到单片机 I/O 口 P1.2

7. 4×4 键盘扩展



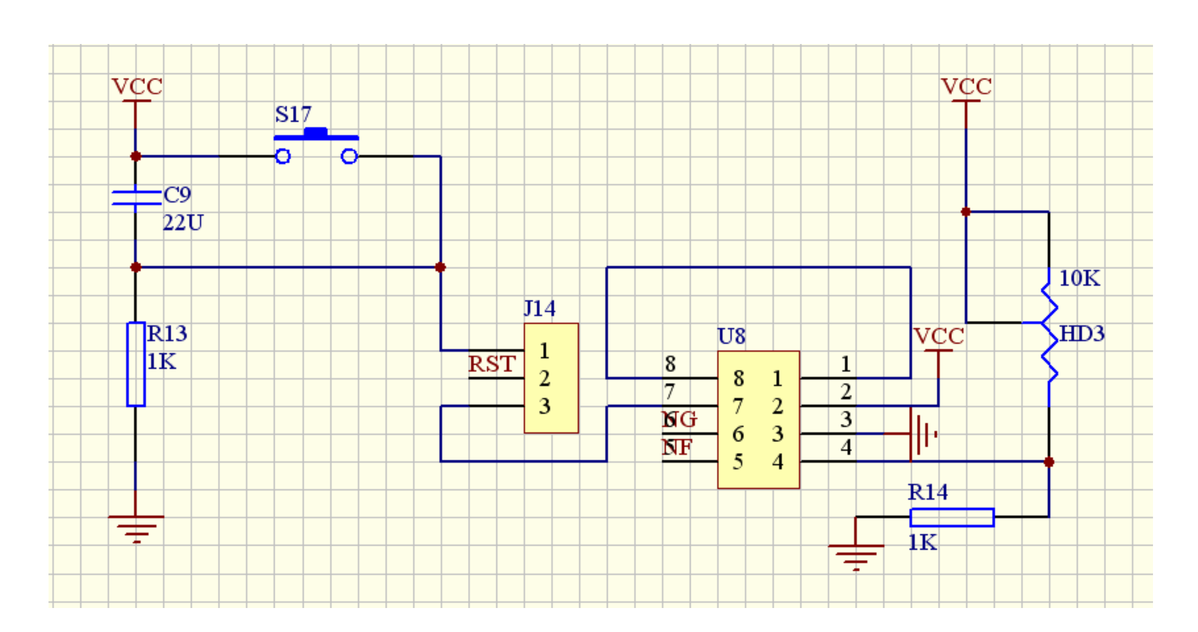
原理:

采用矩阵扫描的形式，识别按键,然后转入相应处理,程序参见光盘部分.

接线说明:

接线柱 J16 接到单片机 P1 口. (J16 的第一个脚对应接 P1.0)，数码管接入.

8. 看门狗+上电自动复位电路+手动复位电路+电源监控



外部看门狗：采用 MAX813L,程序跑飞时可以强制复位,相当于一个定时器,有一个设定的计数值,计数没有溢出之前将其计数值清零,然后重新计数,这个过程即:"喂狗",如果在计数溢出之前没有清零,则计数器溢出,从而产生强制复位信号.工业控制干扰大的场合必用看门狗电路防止出现意外事故.

附带功能:电源检测功能,当电源电压低于门限值时相关引脚电平跳变,可以输入到单片机中断口,使单片机保存数据等. 程序参见光盘部分.

接线说明:

NF 接到 P3.2,NG 接到 P1.4，J14 中将看门狗输出与复位脚用短路冒短接起来。数码管接入.可以试着停止喂狗的语句看数码管显示有何变化.

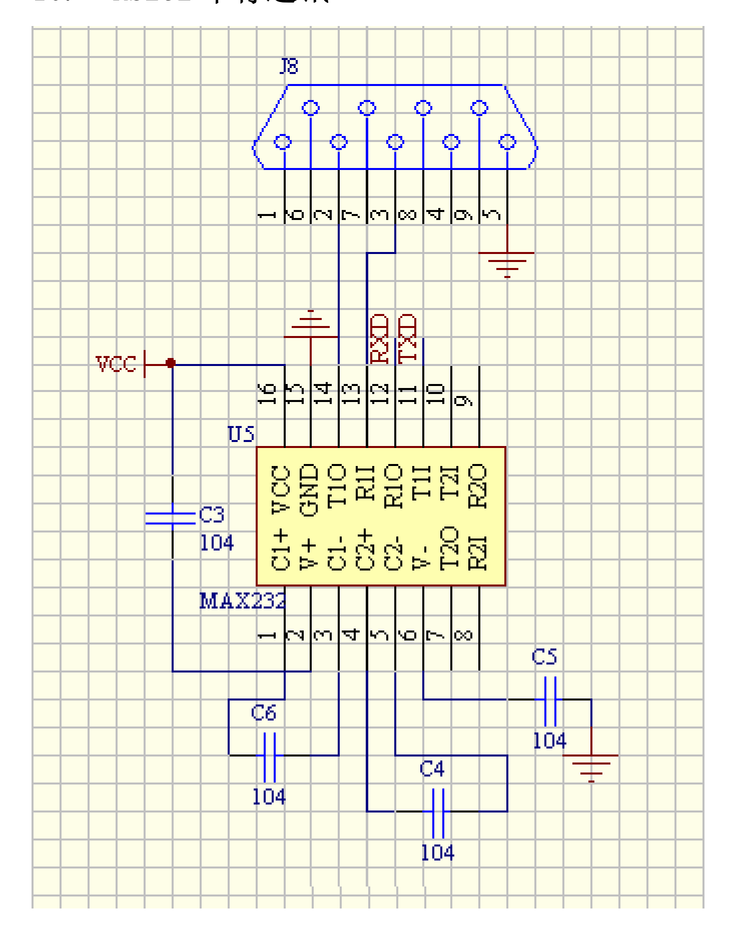
9. 128*64 点阵式液晶显示

所用的液晶型号为 FM12864J（20 个接口） ,使能信号 E 为低电平有效.驱动程序见程序部分. 程序参见光盘部分.

接线说明:

液晶模块用跳线冒接入(即 AA 连接到 P0.0,BB 连接到 P0.1 依次接入,CON1...CON5 分别接到 P2.7……P2.3).

10. RS232 串行通讯



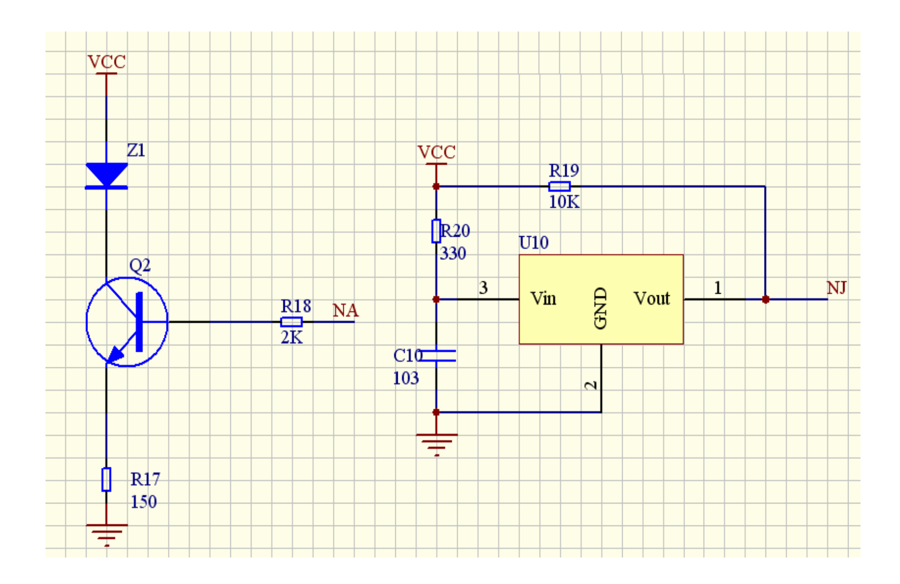
串行通讯可以实现单片机与 PC 机方便的通讯, 51 的输入/输出缓冲器共用一个地址,但是两个不同的寄存器,可以分别用于缓存待发射的和要接收的程序,程序参见光盘部分.

通讯波特率的设定:51 单片机通过定时器中的定时值来设定波特率

接线说明:

开发板上的串口接到 PC 机串口.RXD 接到 P3.0,TXD 接到 P3.1,数码管接入.注意数码管显示的数据为十进制,以十六进制发送 11,则数码管显示为 17.

11. 红外信号发射与接收



红外发射采用红外发射二极管,用软件产生 38KHZ (也可为其他频率的)的载波信号,单片机对信号进行调制然后发射出去,所谓载波调制是非常简单的过程,看看下面的程序即可明白;

红外接收用一体化红外接收头,选用 38KHZ 的(与发射频率一致才能接收),红外头的原理:相当于一个滤波器,过滤频率为特定值的信号,如 38KHZ 等,当没有接收到信号时输出为高电平,接收到特定频率的信号后,把这个特定频率的信号过滤掉,然后输出低电平,至于红外编码的标准,暂可以不涉及,若想用 38KHZ 的遥控器来控制,则要先进行"自学习",即先得到按键对应的编码,在本程序基础上稍加扩展即可写出程序.

常见的两种调制方式 :PWM 方式和 PPM 方式也都是基于这个原理的,发射时先发射起始电平,结果码,然后是地址码,地址反码,数据,数据反码,单片机接收到红外信号然后转入相应处理.下面的程序为简化了的协议,仅发射数据信号.

接线说明:

单片机自发,自收时为:单片机自发射载波,自接收载波,接线:NA 接到 P3.6,NJ 接到 P3.3,单片机 I/O 口 P1^7 接到任一个流水灯 LED 做显示.

单片机对外发射: NA 接到 P3.6 即可(Z1 的封装多了一个"脚",第一,三个为 VCC,中间为低这样红外二极管就可以方便掉头发射).此时为开发板上红外管发射,外部接收.

单片机对外接收: NJ 接到 P3.3 即可,数码管显示接入.外部发射,板上接收头接收.

程序:发射红外信号

```
/*******************************************************************
接收部分：数码管接 P0 口，控制端为 P2.7-P2.4，红外接收头数据口接中断 0
发射部分：接入两个外部中断，2 个按键分别对应两个按键码被发射出去.
整理时间:   2006-10      http://www.ICDev.com.cn
/*******************************************************************
#include <reg52.h>
#define uchar unsigned char
 sbit NA=P3^6;                         //控制红外发射管的亮灭,高电平时发射信号
 int count;                            //延时计数器
```

```
 int endcount;                         //终止延时计数
 uchar flag;                           //红外发送标志
 uchar hongwai_fashe_enable=0;
 uchar k=0x06;
 void    Senddata(uchar p_data);       //发送函数
 void    delay();                      //延时程序,1 秒


void zhongduan1(void)   interrupt 0
{
  k=0x38;
  hongwai_fashe_enable=1;
}

void zhongduan2(void)   interrupt 2
{
  k=0x64;
  hongwai_fashe_enable=1;
}

void main(void)
{
  count = 0;
  flag = 0;
  endcount=0;
  NA=0;                                //
  EA = 1;                              //允许 CPU 中断
  EX0=1;
  EX1=1;
  IT0=1;
  IT1=1;
  TMOD = 0x11;                         //设定时器 0 和 1 为 16 位模式 1
  ET0 = 1;                             //定时器 0 中断允许
  TH0 = 0xff;
  TL0 = 0xe6;                          //设定时值 0 的装入值，输出脉冲频率为 38K，即
26us 中断一次
  TR0 = 1;                             //开始计数

  delay();

  for(;;)
    {
      if( hongwai_fashe_enable==1)
          {
```

```
                Senddata(k);                //开始发送数据
                hongwai_fashe_enable=0;
            }
        }
}



void timeint(void) interrupt 1              //定时器 0 中断处理
{
    TH0=0xff;
    TL0=0xe6;                               //26us 中断一次,再重装数据
    count++;
    if (flag==1)                            //如果允许发送
    {
        NA=~NA;                             //开始红外发射管发射载波信号（不停的反相），接
收时认做低电平
    }
    else
    {
        NA = 0;                             //没有发射载波信号
    }
}


void Senddata(uchar p_data)                 //发送函数
{
    int i;
    uchar getdata;                          //操作的数据

//发送 9ms 的起始码（发射载波），9000us,为 26us 的 346 倍
    endcount=346;
    flag=1;                                 //允许发送载波
    count=0;
    do{}while(count<endcount);
//发送 4.5ms 的结果码（红外管不发光，接收时认做高电平）
    flag=0;
    endcount=173;                           //红外管发射信号
    count=0;
    do{}while(count<endcount);

//发送八位数据
    getdata=p_data;
    for(i=0;i<8;i++)
```

```
  {
     if(getdata-(getdata/2)*2)            //取出每一位的信息
     {
       endcount=65;
     }
     else
     {
       endcount=21;
     }
     flag=1;
     count=0;
     do{}while(count<endcount);
     flag=0;

     endcount=22;                         //发出高电平
     count=0;
     do{}while(count<endcount);
     getdata=getdata>>1;
  }
  flag=0;
  NA=0;

}


void delay()                              //延时程序，大约 1s
{
  int i,j;
  for(i=0;i<1000;i++)
  {
    for(j=0;j<125;j++){;}
  }
}
```

对应接收程序:

```
/*****************************************************************************
函数功能:  利用单片机通过 38K 载波调制发射红外信号
接线说明:  见上面说明
程序说明:  单片机自发自收时,只能自发射载波信号,自接收到载波则接收头电平跳变;
           可用另一个单片机组成最小系统,发射红外信号.
整理时间:  2006-10        http://www.ICDev.com.cn
*****************************************************************************/
```

```
#include <reg52.h>
#define uchar unsigned char
char shuju[10]={0x5f,0x44,0x9d,0xd5,0xc6,0xd3,0xdb,0x45,0xdf,0xd7}; //数码管显示数字0－9
 int   hongwai_receiver[9]={0,0,0,0,0,0,0,0,0};        //存放载波周期数
 uchar   hongwai_re_cishu=0;                            //
 int   hongwai_shuju=0;                                 //载波周期的个数
 uchar   hongw_receive=0;                               //存放接收到的数据
bit      hongwai_re_enable=0;                           //允许计算低电平的维持时间
bit      hongwai_dis_enable=0;                          //允许显示接收倒的数据
sbit     hongwai_get=P3^3;                              //用于读端口操作
void     control();                                     //对接受的数据进行识别处理

void hongwai_Delay(int j)                               //延时函数
{
       int m;
       for(m=0;m<j;m++)
       {
              int i;
              for(i=0; i<300; i++){;}                   //约 1ms
       }
}


void Led(int k)
{
   if(k>999)
   {
    P2=P2&0x7f;                                         //P2.7 输出低电平，选通千位数
    P0=shuju[k/1000];                                   //取千位数
    hongwai_Delay(2);                                   //延时
    P2=P2|0xff;                                         //销隐
   }
   if(k>99)
   {
    P2=P2&0xbf;                                         //P2.6 输出低电平，选通百位数
    P0=shuju[k%1000/100]+0x04;                          //取出百位数,点亮数码管百位后的那一点
    hongwai_Delay(2);                                   //延时
    P2=P2|0xff;                                         //销隐
   }
   if(k>9)
   {
    P2=P2&0xdf;                                         //P2.5 输出低电平，选通十位数
    P0=shuju[k%100/10];                                 //取十位数
```

```
    hongwai_Delay(2);                        //延时
    P2=P2|0xff;                              //销隐
  }
  if(k>=0)
  {
    P2=P2&0xef;
    P0=shuju[k%10];                          //取出个位数
    hongwai_Delay(2);
    P2=P2|0xff;
  }
}

void waibu_int(void) interrupt 2
{
    EX1=0;
    hongwai_re_enable=1;
}


void main(void)
{
  IT1 = 1;                  //设外部中断 1 为下降沿触发
  EX1=1;
  TMOD = 0x11;              //设定时器 0 和 1 为 16 位模式 1
  ET0 = 1;                  //定时器 0 中断允许
  EA = 1;                   //允许 CPU 中断
  TH0 = 0xff;
  TL0 = 0xe6;               //设定时值 0 的装入值，输出脉冲频率为 38K，即 26us 中断一次
  TR0 = 1;                  //开始计数
  P0=15;
  while(1)
   {
      control();
   }
}
void timeint(void) interrupt 1                         //定时器 0 中断处理
{
  TH0=0xff;
  TL0=0xe6;                                            //26us 中断一次,再重装数据
  if(hongwai_re_enable==1)
  {
        if(hongwai_get==0)
          {
            hongwai_shuju++;                           //计算 0 电平维持的时间
```

```
        }
      else
        {
          EX1=1;
          hongwai_re_enable=0;
          hongwai_receiver[hongwai_re_cishu]=hongwai_shuju;
          hongwai_shuju=0;
          hongwai_re_cishu++;
        }
      if(hongwai_re_cishu==9)                    //接收完成
        {
          hongwai_re_cishu=0;
          hongwai_dis_enable=1;                  //允许显示接收的数据
        }
  }
}


//发送数据时先发射最低位
void control()
{
   int i;

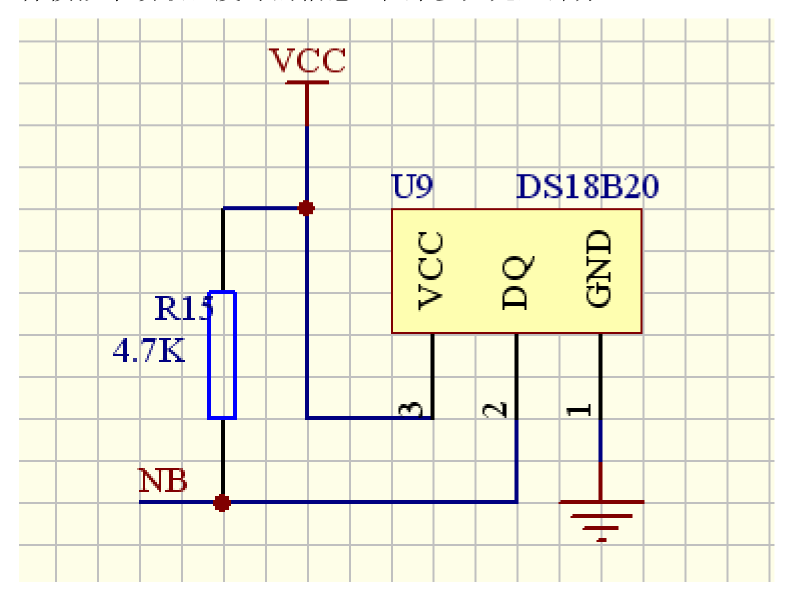
   if(hongwai_dis_enable==1)                     //
    {

        if(300<hongwai_receiver[0]<360)          //检测起始码是否正确 300
          {

            for(i=1;i<9;i++)
               {
                 hongw_receive>>=1;              //数据右移
                 if((hongwai_receiver[i]<80)&&(hongwai_receiver[i]>40)) //是否高电平
                    { hongw_receive|=0x80;    }
               }
          }
        hongwai_dis_enable=0;                    //
    }
  if(hongw_receive==0x38)hongw_receive=11;
  else if(hongw_receive==0x64) hongw_receive=68;
  Led(hongw_receive);                            //控制显示数据已经接收
}
```

12. 温度测量

温度测量模块采用芯片 DS18B20,为单总线接口的器件,51 单片机没有单总线接口,故用软件模拟来读取温度计的信息. 程序参见光盘部分.



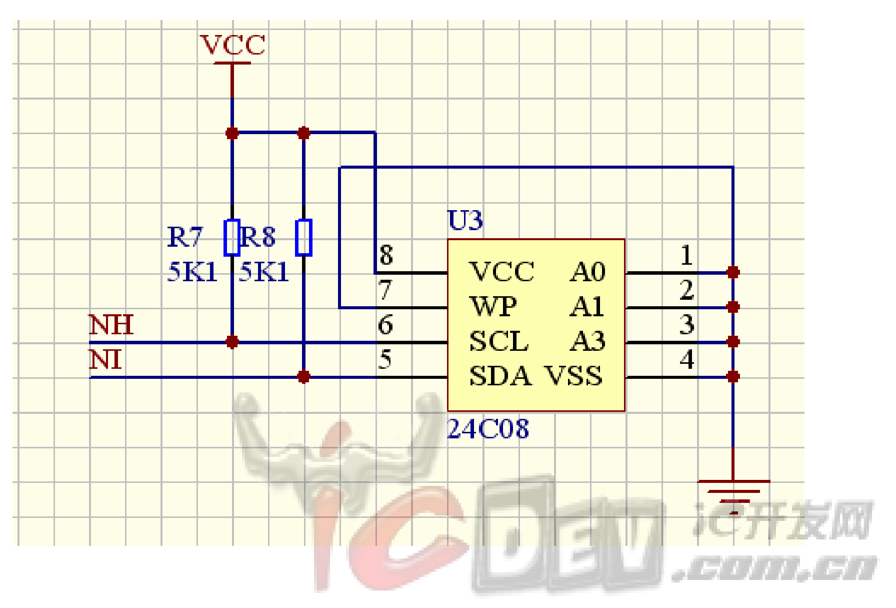
接线说明:

NB 接到单片机口线 P3.7，数码管接入.

13. I2C 接口的 E2PROM

AT24C08 为 EEPROM 芯片,I2C 总线的接口形式,具有接口简单,扩展方便的特点,多个 I2C 接口的芯片可以同时挂到总线上,不会相互冲突,节约单片机口线.

51 单片机没有 I2C 总线的接口,只有通过软件来模拟,编程只须按照协议的时序要求即可. 程序参见光盘部分.

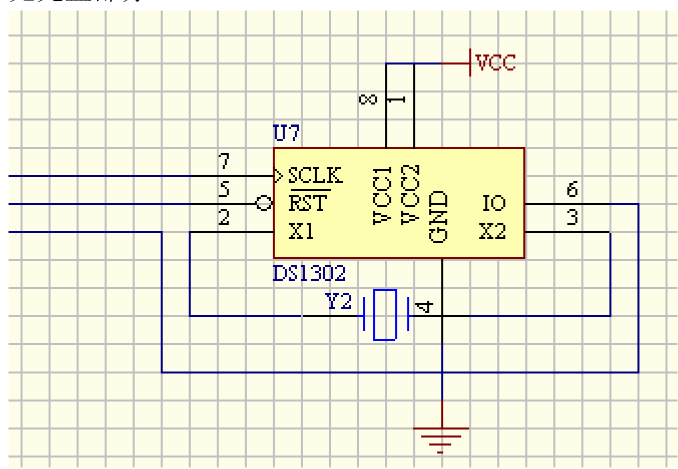


接线说明:

NH,NI 分别接到 P1.0 和 P1.1,数码管接入.

14. 时钟芯片

设计中时钟芯片采用串行时钟芯片 DS1302, 具有占用 I/O 口线少的特点,可以外接蓄电池作后备电源,可以对蓄电池涓流充电.通过三根线与单片机相连,可以读写时钟信号. 程序参见光盘部分.



接线说明:

将 DS1302 三根线接到单片机口线 P2.2,P2.1 和 P2.0,数码管接入.

15. PS2 接口

键盘接入原理:

PC 机使用的通用键盘,用到四根线,一根电源线,一根地线,一根数据线和一根时钟线.键盘能够自动产生按键对应的键码(8 位数据),如按下小写字母 a,则键盘能够将这一动作转换为数据 0x38 .单片机读取按键动作所对应的键值,只需要与键盘输出时序一致即可.

键盘输出数据时键盘的时钟线作中断信号,数据线传输数据;单片机在中断服务程序中读取数据线上面的信号即可. 程序参见光盘部分.

接线说明:

NE 接到单片机口线 P3.4,数码管接入系统作显示. 本程序中只定义了数字键 0-9,按下后数码管立即显示按键值.

（Ⅱ）综合实验

综合应用的实例会不断补充,请关注网站的最新信息.

16. 可调式电子万年历

功能描述:

输入: 红外发射信号/PS/2 接口接入键盘

输出: 液晶显示,讯响器发声.

功能: 显示万年历信息(年月日,星期,当前时间);当前环境的温度;定时闹铃;倒计时;屏幕下方文字信息动态显示.

PC 机键盘接入按键定义: 小写 a:开/关闹铃　　b:开/关倒计时

红外发射(需自行搭建一个最小系统,需要红外发射电路,需要两个按键引入外部中断 0,中断 1,即:与开发板上的电路一致,程序中发射控制端为:P3.6),发射程序在实验 16 的程序文件夹中.

接线说明:

接入液晶显示屏,DS1302,DS18B20,音乐演奏电路,PS/2 口,红外接收,上电自动复位,如需要通过串口控制则接入串口实验的线路等.

显示界面如下:



设计原理及程序见光盘.